Viewpoint-Independent Object Class Detection using 3D Feature Maps

Jörg Liebelt, Cordelia Schmid, Klaus Schertler

Viewpoint-Independent Object Class Detection using 3D Feature Maps

Joerg Liebelt
IW-SI, EADS Innovation Works
D-81663 Munich, Germany
joerg.liebelt@eads.net

Cordelia Schmid
LEAR, INRIA Grenoble
F-38330 Montbonnot, France
cordelia.schmid@inrialpes.fr

Klaus Schertler
IW-SI, EADS Innovation Works
D-81663 Munich, Germany
klaus.schertler@eads.net

Abstract

This paper presents a 3D approach to multi-view object class detection. Most existing approaches recognize object classes for a particular viewpoint or combine classifiers for a few discrete views. We propose instead to build 3D representations of object classes which allow to handle viewpoint changes and *intra-class variability. Our approach extracts a set of pose and class discriminant features from synthetic 3D object models using a filtering procedure, evaluates their suitability for matching to real image data and represents them by their appearance and 3D position. We term these representations* 3D Feature Maps*. For recognizing an object class in an image we match the synthetic descriptors to the real ones in a 3D voting scheme. Geometric coherence is reinforced by means of a robust pose estimation which yields a 3D bounding box in addition to the 2D localization. The precision of the 3D pose estimation is evaluated on a set of images of a calibrated scene. The 2D localization is evaluated on the PASCAL 2006 dataset for motorbikes and cars, showing that its performance can compete with state-of-the-art 2D object detectors.*

1. Introduction

Existing work on object detection based on local features can be roughly separated into two groups, i.e., detection of specific objects and of object classes. Numerous approaches propose solutions for viewpoint-independent detection of specific objects and significant progress has been made recently [11, 16, 18]. In contrast, methods for detecting generic object classes have to handle significant intra-class variations in addition to multiple viewpoints. Most existing approaches represent multiple viewpoints either implicitly, i.e., one detector is trained on images taken from different viewpoints, or use a set of viewpoint specific detectors and combine the output [4, 9, 21]. Very recently more sophisticated methods make use of a 3D model structure [19, 24]. However, even these approaches only determine 2D regions of interest in the image plane as the localization output. Yet, in many cases a 3D pose estimation of the detected generic object would be useful.

In the present work, we describe an approach to viewpoint-independent object class detection which does provide information on the 3D pose of the detected object. Unlike most recent approaches, we do not build a model from 2D features and their geometric constraints, but resort to a database of existing, fully textured synthetic 3D models. Our approach computes a 3D representation for each object category, thereby facilitating viewpoint-independent recognition. The local features obtained from rendered synthetic objects have to be selected during training in order to be suitable for a reliable matching to real image features (section 3). During detection (section 4), local features from real images are matched to the synthetically trained ones; the experimental results show the pertinence of the approach. Each match casts votes to determine the most likely class and 3D pose of the detected generic object. The most promising votes are then evaluated and refined with respect to their geometric consistence with the 3D model using a robust pose estimation step (section 5). In section 6, we present experimental results on the 2006 PASCAL datasets for motorbike and car models [3] and analyze the precision of the 3D pose estimation using a calibrated scenario.

2. Related Work

Research on viewpoint-independent object class recognition can be roughly separated into two categories. On the one hand, several approaches dynamically build an approximate 3D representation from 2D training data in order to achieve some degree of viewpoint invariance [9, 19, 21]. On the other hand, the use of existing 3D models has been advocated in the past [5, 13, 17, 20, 23] and remains an appealing strategy [24].

Dynamically built representations for viewpoint-independent object recognition have been proposed in the field of face detection, where several authors deal with multiple viewpoints by combining the results of separate single-view detectors ([4, 22]). For general object categories, Thomas et al. [21] suggest linking Implicit Shape Models for specific viewpoints amongst each other, thereby achieving a detection over multiple viewpoints at the cost of an expensive training process on manually

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

Figure 1. Examples for 3D models of our two-class training database.

segmented viewpoint-specific examples. Kushal et al. [9] enforce local geometric constraints between Partial Shape Models which are dense locally rigid assemblies of image features, thereby achieving robustness to viewpoint changes as well as better 2D localization performance. Similarly, Savarese et al. [19] determine homographies of groups of local features in order to map large 2D image regions onto a collection of near-planar parts to form a viewpoint-independent 3D model. Note that most of the existing approaches either need 2D training data with manual viewpoint annotations or they constrain the 3D representation of the trained objects to a collection of planar subparts.

An alternative lies in using a database of existing 3D CAD models, given their ubiquitous availability and increasing realism in recent years. In the late 1980s and 1990s, this idea has already been advocated by several researchers. Some authors resorted to flexibly aligning groups of consistent edge segments by probabilistic matching [13] or relational graphs [5]. Others performed geometric indexing based on invariants, typically curvature or edges, which cast votes into a hash table of potential object poses [17, 20, 23]. Many approaches include an additional step to verify and refine the geometric consistency of the most likely hypotheses [7]. The majority of these solutions rely on geometric primitives which are in general not sufficiently robust and discriminant for generic object categories. Recently, Khan et al. [24] addressed this issue by collecting patches from 2D images with 3D viewpoint annotations and mapping these patches onto an existing 3D CAD model. However, the suggested texture mapping of 2D patches onto a single 3D model is prone to cause artifacts in the appearance representation. Moreover, a single 3D model is in general not sufficient to capture the geometric variations within an object category.

In this work, we build on the idea of using existing 3D CAD models. However, we do not rely on geometric features for matching, but use a vocabulary of photometric descriptors which are more discriminant and are shown to match to real images. In addition, the available 3D geometry of the object category models is systematically exploited to achieve viewpoint-independent state-of-the-art 2D object recognition results and an approximate 3D pose prediction.

3. Training

3.1. Training Data

Training an object detector from real images implies a number of difficulties, such as varying imaging conditions and presence of background clutter. Several authors have proposed solutions to these issues, for example Marszalek and Schmid [14] suggest a spatial weighting to segment objects from the background. Training on synthetic images rendered from 3D models represents an alternative solution which guarantees training data with perfect object segmentation.

As mentioned in the previous section, most authors of recent publications have preferred to build partial or full 3D models from 2D training data, often based on complex structure-from-motion techniques. However, the growing importance of so-called virtual reality applications and their need for 3D representations of real-world objects have resulted in a near-ubiquitous availability of 3D models for many categories with realistic textures and dimensions. Figure 1 shows a few examples for synthetic 3D models of our training database. By choosing to train our detector on rendered views of synthetic 3D models, we circumvent both instable training conditions and complex model-building at the cost of a possibly reduced descriptor similarity.

3.2. Local Features

Our approach is based on local features which are extracted from rendered views of synthetic 3D models. The performance of our object class detector, as is the case with any other feature-based detector, depends heavily on descriptor similarity. In our case, it depends on the ability to establish correspondences between descriptors extracted from synthetic images and from real images.

Here, we use the FastHessian feature detector in combination with the SURF descriptor [1]. Experimental results show that this image description allows to match synthetic with real images when combined with a discriminative filtering step, see section 6. Unlike the standard parameters suggested by [1], our detection uses a smaller sampling step of one pixel in order to also capture small objects in

the images, and the descriptor is used in the extended 128-dimensional upright (i.e., not rotation invariant) version.

3.3. Model Acquisition

A truly viewpoint-independent object detector would require training an object representation which continuously covers the entire camera view sphere. To reduce the complexity of the problem, we resort to a discrete representation where gaps in the view sphere are bridged by the invariance of the local features.

First, all our 3D models are scaled to fit into a unit bounding sphere and they are oriented along their dominant dimension. For each model, its minimum-volume enclosing rectangular bounding box is computed from its mesh geometry. To further simplify the problem, we determine the average ratio of the bounding box dimensions (length, width and height) within each object class, resulting in a single scale parameter to represent a class-specific bounding box. Each model is now rendered from a discrete number of viewpoints, characterized by the following conditions as depicted in figure 2:

1. The camera is positioned at (azimuth, elevation, distance), looking at the world coordinate origin at (0,0,0).
2. The model is rendered with its bounding box centered at the origin, called the *0-pose*.
3. The distance between object and camera is varied at discrete steps in order to account for the fact that the scale invariance of the features is limited in practice.
4. The orientation parameters azimuth and elevation run through a set of discrete values.

The choice of the discretization has to take into consideration the degree of invariance offered by the chosen local feature descriptors as well as the viewpoints which are to be encountered by the detector when working on real images. See section 6 for the specific discretization used in our experiments.

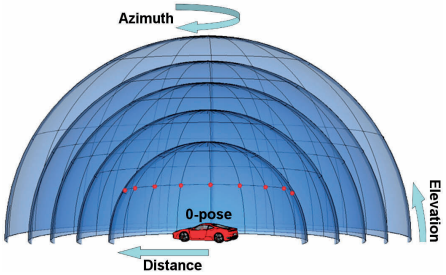

Figure 2. Discretization of the camera parameters azimuth, elevation and distance during training.

A distinct bounding box pose for a given viewpoint can now be described by the three-dimensional parameter set $\lambda = (a, e, d)$ with a, e, d being azimuth, elevation and distance of the camera pose (directed per definition towards the world origin). Each distinct set of parameters is assigned an identification number which will be used during the voting step. Together with the object class category cat of the training model, the parameter set λ forms the *object hypothesis*. These pose hypotheses are all relative to the internal virtual camera calibration matrix K_v chosen for the rendering step. As a consequence, the 3D pose estimation provided by our method will be relative to the same internal camera parameters; our method only provides an estimation of the external camera matrix V_v.

For each object hypothesis, we then collect local features. Each feature is annotated with the identification number of the object hypothesis which describes its originating viewpoint and bounding box along with a weight which corresponds to the inverse of the number of features found under this viewpoint. This weight is necessary to balance each viewpoint's contribution, since for example profile views of an object typically cover a larger image surface and therefore result in significantly more features than frontal views. In addition, each feature stores its 3D position relative to the model geometry in the normalized object coordinate system. This 3D position, which corresponds to the feature location in the image after backprojection onto the object geometry, allows for a 3D pose estimation. We term these groups of 3D-annotated features *3D Feature Maps*, since they contain all the information necessary to roughly reconstruct in 3D the object hypothesis from which they originated.

3.4. Discriminative Filtering

The local features should be discriminant with respect to the object category and each discrete viewpoint. At the same time, the features have to be invariant towards small local pose variations in order to bridge the gaps between the discretely sampled training viewpoints; moreover, they have to be robust in the presence of background. This can be achieved by a discriminative filtering procedure similar in spirit to the method of [11]. In section 6, we show the importance of the discriminative filtering for the detection performance. Filtering consists of the following steps, see figure 3 for an illustration:

1. Each training object is rendered once with the exact viewpoint parameters in front of a white background; local features are collected for the rendered image, in the following identified as the *default feature set*.
2. The training object undergoes a sequence of slight variations of each of the three pose parameters, typically covering $\pm 1\%$ of its respective parameter space, leading to 3^3 evaluated parameter combinations.
3. For each pose variation, the object is rendered three times, in front of a white, a real and a synthetic background (see figure 3).
4. Each pose variation and background yields local features which are matched to the *default feature set* w.r.t. descriptor distance and 3D position distance af-

ter backprojection into 3D world coordinates. The features of the default set are weighted according to the number of matches for each pose and background variation, thus giving a higher importance to the more discriminant and robust ones.

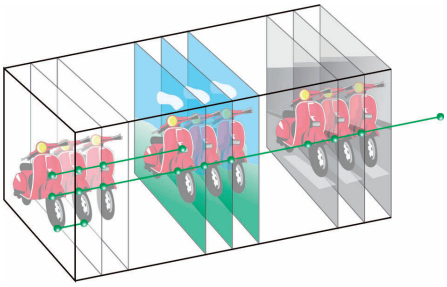

Figure 3. Discriminative filtering of the features during training. The features are weighted according to their stability w.r.t. different backgrounds and small local pose variations.

The discriminative filtering could in principle be extended in the same way to more rendering parameters, such as lighting and imaging conditions.

To reduce the complexity of the approach during matching to real images where a geometry-based filtering is not available, the number of features has to be reduced as early as possible in the processing chain. We, therefore, train a simple two-class SVM classifier with a radial basis function kernel on the synthetic object features harvested during training and a real background feature set in order to differentiate between relevant object and irrelevant background descriptors. On average, up to 60% of the real image features are discarded by the SVM classification.

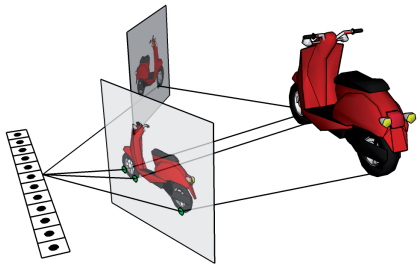

Figure 4. Each codebook entry stores the mean descriptor and the 3D positions of all the similar features which form a cluster.

3.5. Codebook Construction

Similar to many existing approaches [6] we construct a visual codebook of k elements by clustering the harvested discriminant descriptors with the k-means algorithm.

Each cluster stores a list of the discrete poses of the features which contributed to this cluster, along with their viewpoint-specific weights (see section 3.3). From the 3D positions of the training features merged into a single codebook cluster, we build a data structure consisting of multiple linked lists. This data structure allows to quickly recover all potentially visible 3D feature positions of a given cluster under a given viewpoint. This information is later used as input for the geometry verification. Figure 4 shows an example cluster with the links to the 3D positions of its contributing features.

Additionally, in order to accommodate for the presence of background in real images, local features are harvested from a real background image set, and clustered into a separate codebook in the same way as for the other object classes. Unlike object features, background features do not carry 3D information, but a void ID to indicate that they can be discarded during detection. The initial SVM-based feature pre-classification does not supersede this much more fine-grained background class.

4. Detection

During detection, local features are extracted from the image and matched with at most the n closest codebook entries, provided that the matches fulfill the nearest neighbor distance ratio (*NNDR*) criterion ([15]); in our experiments, we use $n = 5$. Next, votes are cast for the respective pose parameters. Each cluster casts votes for the voting bins of the discrete poses contained in its internal list. The interpretation of the voting result can be expressed analogously to the work of [10] as follows:

Each extracted feature descriptor f corresponds to a codebook entry c_j with probability

$$p(c_j|f) = 1.0 - e^{-(d_b(f)/d(f,c_j)-1)} \quad (1)$$

where $d(f, c_j)$ is the descriptor distance of f and c_j and $d_b(f)$ is the distance of f to the closest cluster belonging to a different object class than c_j. For each codebook entry, we can derive the distribution of the parameter sets $\lambda = (a, e, d)$ and the classification $cat = \{object, background\}$ from the training data as $P(cat, \lambda|c_j)p(c_j|f)$. The vote of each match in favour of an object hypothesis consisting of (cat, λ) then has the weight

$$p(cat, \lambda|f) = \sum_{j=1}^{n} P(cat, \lambda|c_j)p(c_j|f). \quad (2)$$

Note that the 2D location of the extracted feature does not contribute to the voting weight. It is only used in the subsequent geometry verification step for a 2D-3D pose refinement.

Figure 5 visualizes the result of voting within one class-specific voting space. The bin with the maximum sum of votes over all features indicates the most likely object hypothesis (cat, λ) in the *0-pose* (see section 3.3). We perform a non-maxima suppression within the voting space and retain the maximum votes as the potential pose hypotheses. Note that the voting cannot distinguish between symmetric orientations which yield nearly identical feature distributions. These ambiguities have to be resolved in the pose refinement step described in the next section.

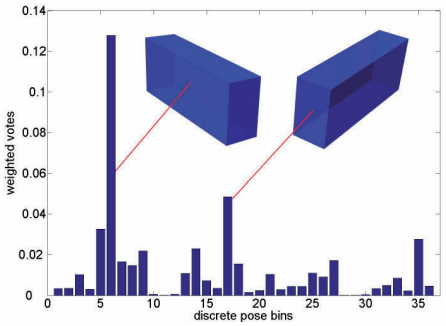

Figure 5. Histogram of the votes cast by the matched features into the discretized pose bins. The bounding boxes illustrate the poses corresponding to the two local vote maxima. Symmetric object orientations yield similar features. For simplification, only the azimuth pose bins are shown.

5. Pose Refinement

Due to incorrect descriptor matches, ambiguous symmetric poses, background clutter and limited feature invariance, the pose hypotheses typically contain geometrically inconsistent results as well as overlapping or multiple detections of the same object.

To separate the correct from the inconsistent hypotheses, we perform a pose estimation to determine the number of matches which are consistent with the model geometry. Based on the pose hypothesis identified in the voting step, for each matched codebook entry we recover the list of all 3D positions visible under this pose hypothesis. Each of these 3D positions, along with the descriptor of the cluster it belongs to, forms a potential *model feature*. Matching pairs of model and image features are then sorted according to their descriptor distances and fed into a RANSAC loop. Inside the RANSAC loop, on each subset of three 3D-2D model-image feature pairs a perspective three-point (P3P) method [8] estimates the extrinsic camera parameters which project the model features onto the image features. The P3P is based on the intrinsic parameters of the virtual camera which have been used to render the models in the training step. For each pose estimated from a subset of three feature pairs, the number of matching inliers among all features is determined. An inlier is defined as a pair of 3D model and 2D image features which are close in descriptor and position space after the model feature has been projected into the image plane. The pose estimation with the maximum number of inliers is retained.

Since the chosen 3D representation contains all 3D locations under which a given feature cluster was found during training, the geometry matching can accommodate for variations of the object geometry. Note that we perform neither a non-rigid registration of the model nor an iterative optimization. Instead, the success of the closed-form pose estimation depends on the fact that among all the feature positions identified during training, a minimum of 4 corresponding, geometrically consistent feature positions can be discovered in the input image. Features occurring at positions which have not been trained, cannot be matched either.

The pose refinement allows for the detection of multiple object instances present in an image, since each of the locally maximal hypothesis votes will be evaluated. For object geometries similar to those of the training models, an object instance will typically yield a single refined hypothesis. In case of significant deviations from the trained geometric configurations, a single object might result in several pose estimations of its subparts, differing only slightly in translation and scale; our method detects these cases and combines them into a single hypothesis with an extended bounding box. Occlusions are handled implicitly: as long as the visible part of the object yields enough geometrically consistent feature matches, the correct object pose will be found; see section 6.2, figure 7, right, for an example.

6. Evaluation

6.1. Dataset and Evaluation Criteria

For training, we used 8 synthetic models for the class "motorbike" and 50 synthetic models for the class "car". The models come from different free and commercial CAD model databases, notably turbosquid.com, 3d02.com and doschdesign.com. The experiments were performed using these two classes and an additional background dataset. As background dataset we used the car dataset and annotations of the PASCAL 2006 training data with the object annotation masks cut out. The codebook contains $K = 2000$ clusters per class as described in section 3.5.

Table 1 summarizes the parameter space discretization used for training. We have experimentally found these values to best cover the viewpoints and object poses, given the invariance of the descriptors used. Both increasing and decreasing resolution resulted in a loss in performance due to less pronounced maxima or less precise pose estimations. Note that for our experiments, we have chosen to include the model scale variation into the camera pose estimation as outlined in section 5.

Table 1. Choice of discretization parameters for training

description	values
azimuth a	$0..360^\circ$ in 10° steps
elevation e	$0..40^\circ$ in 20° steps
object distance d	$4.5, 6, 7.5$ [units]

In order to evaluate the performance of our detector w.r.t. 2D ground truth bounding boxes, we use the detection quality criterion suggested by [3]: For a correct localization, the overlap a_o between predicted bounding box B_p and ground truth bounding box B_{gt} must exceed 50% as defined by

$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}. \quad (3)$$

Our 2D localization is created by projecting the 3D bounding box into the image plane and computing the convex hull

of the 2D projection of the bounding box corners which is a better approximation than just a rectangular 2D bounding box.

6.2. 2D Localization

In the following, we present the evaluation results of our detector on the PASCAL 2006 test set [3]. The evaluation follows the conventions of the PASCAL 2006 object detection challenge.

Figure 6 shows precision/recall curves for the PASCAL VOC2006 car (figure 6, left) and motorbike (6, right) test datasets; we evaluated on the entire test sets, each consisting of 2686 images. Our approach produces few false positives and achieves an excellent detection precision as long as sufficient feature matches for an accurate pose estimation can be found. No fallback 2D detectors are used when the minimum required number of 4 geometrically consistent matches necessary for a 3D pose estimation can no longer be found; consequently, recall is lower than for pure 2D detectors. In figure 6, we provide the results of the 2D detector of [2] which performed best on the 2006 dataset in the PASCAL Challenge 2007. Their method achieves a higher average precision due to a better recall behavior; however, on the motorbike test set, our method detects fewer false positives due to the more restrictive geometry verification and consequently maintains a detection precision above the results of [2] for a significant fraction of the test set. On the motorbike dataset, our approach performs better than on the car dataset. Car objects usually have less textured structure which reduces the number of pose-discriminant features found during training and matching. Since the work of [24] is similar to our approach in that they resort to an existing 3D model geometry, we also show their results on the VOC2006 motorbike test set in figure 6. Although they use real training images and directly focus on 2D localizations, their P/R curve is lower. This might be due to the much smaller number of images used in their training procedure.

Figures 7, 8 shows some examples of successful detections on the PASCAL 2006 test set. The 2D detections and the estimated 3D poses are visualized using semitransparent 3D bounding boxes, backprojected onto the image plane. Figure 9 depicts some failed detections. The failed detections are due to incorrectly established feature correspondences, failed 3D pose estimations due to ambiguities or geometrically inconsistent feature layouts, object geometries which have not been trained for, or an insufficient number of features in case of small objects.

It should be noted that detection performance vitally depends on descriptor similarity. Since our training and testing descriptors stem from different data types (synthetic resp. real images), the overall descriptor similarity is reduced, resulting in fewer correct matches. As a consequence, the discriminative filtering step outlined in section 3.4 is crucial in achieving a sufficient matching performance.

If the discriminative filtering step during training was omitted, training would result in a codebook which has little discriminativity with respect to object/background separation, pose resolution and 3D position stability. On average, more than 90% of the features are discarded during filtering as being not sufficiently stable and discriminant. We found that the average precision on the same dataset falls from 0.453 to 0.125 for motorbikes and from 0.363 to 0.1 for cars when not using discriminative filtering.

6.3. 3D Pose Estimation

In addition to the 2D localization, the proposed approach yields an estimate of the generic object's 3D pose. In order to evaluate the precision of the 3D pose estimation, we took a set of images of two toy cars with a calibrated camera; see figure 10 for an example. For each image, the actual 3D world coordinates of the toy cars (x^r_{gt}) w.r.t. the calibration pattern as well as the intrinsic (K_r) and extrinsic (V_r) camera matrices are known. Given the intrinsic calibration matrix of the virtual camera used during training (K_v), we map the virtual 3D bounding box world coordinates (x^v_{est}) (in homogeneous notation) with the estimated extrinsic matrix of the virtual camera (V_v) into the world coordinate system of the calibrated real scene and evaluate the error relative to measured ground truth (x^r_{gt}):

$$x^r_{est} = V_r^{-1}\, K_r^{-1}\, K_v\, V_v\, x^v_{est}\ . \tag{4}$$

We normalize the homogeneous component to account for the scale factor.

Table 2 lists the errors of the 3D estimations over 14 calibrated poses of toy cars. The position error is measured as the Euclidian distance between the centroids of the ground truth and the estimated bounding boxes, while the orientation error is measured as the angle between their dominant axes. Although the precision of our pose estimation cannot compete with methods for registration or tracking of a specific model (cf. [12]), it is sufficient as an initialization for these methods. The position error is mainly due to the underestimation of the distance between object and camera. This behavior is probably caused by our choice of using a constant scale factor and constant bounding box dimensions for all objects and resolving the scale estimation via the camera pose.

Table 2. Evaluation of the 3D pose estimation for a calibrated scene; 14 ground truth experiments with two toy cars (cf. toy car length $280\,mm$).

	mean	std. dev.
position error of bounding box centroid	$33.9\,mm$	$21.74\,mm$
angular error of main bounding box axis	$10.7°$	$5.2°$

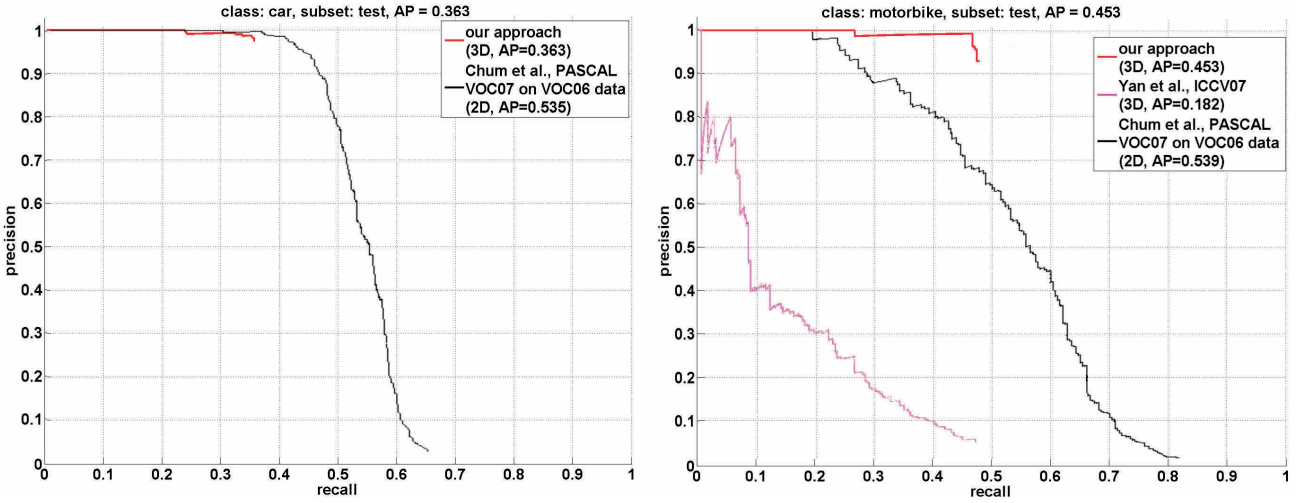

Figure 6. Precision/Recall for the PASCAL 2006 car (left) and motorbike (right) dataset of our approach, the 3D approach of [24] and the best PASCAL Challenge 2007 detection on the 2006 test set.

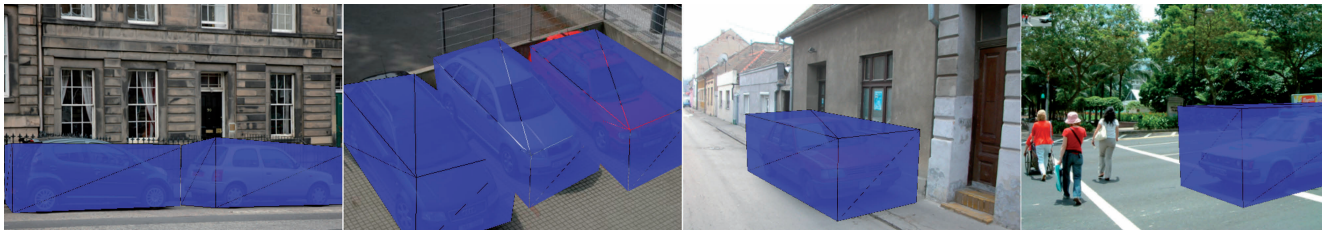

Figure 7. Some successful 2D detections from the PASCAL 2006 car test set. This figure is best viewed in color.

Figure 8. Some successful 2D detections from the PASCAL 2006 motorbike test set. This figure is best viewed in color.

Figure 9. Remaining issues illustrated on a few examples from the PASCAL 06 dataset (from left to right): incorrect feature matches, failed 3D pose estimation, unknown object geometry, not enough features for a successful pose estimation due to the size of the objects. This figure is best viewed in color.

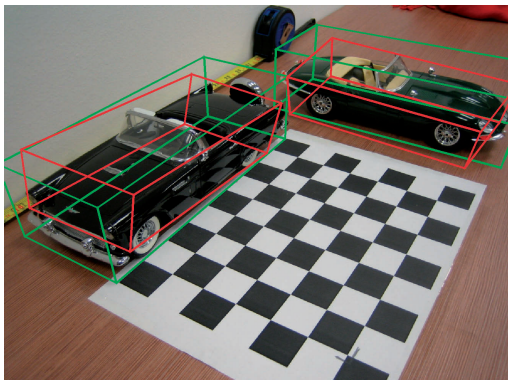

Figure 10. Calibrated scene used for 3D evaluation. Measured ground truth bounding boxes are displayed in green, estimated bounding boxes in red. (Errors pos./orient.: left car $21.9\,mm/6.55°$, right car $60.3\,mm/6.14°$).

7. Conclusion

We have presented a new approach to viewpoint-independent object class detection. The main contributions of this work lie in the training process of local 3D-aware features from synthetic 3D models, the selection of pose- and class-discriminant descriptors and the extension of the traditional probabilistic voting scheme to 3D (i.e., beyond 2D interest regions). The method generates an approximate 3D pose hypothesis for generic object classes which is then refined by a full 2D-3D pose estimation. Future work will focus on extending our approach to more object categories, including rigid as well as non-rigid geometries, and on replacing the direct pose estimation with an iterative optimization scheme. To improve the recall of our approach, we will investigate the use of dense features and analyze the effect of augmenting the synthetic descriptor codebook with real descriptors.

Acknowledgments

J. Liebelt and K. Schertler acknowledge support from EADS MAS and EADS DE.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision*, 2006.

[2] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[3] M. Everingham, A. Zisserman, C. Williams, and L. V. Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) results. Technical report, University of Oxford, University of Edinburgh, KU Leuven, 2006.

[4] Z.-G. Fan and B.-L. Lu. Fast recognition of multi-view faces with feature selection. In *International Conference on Computer Vision*, 2005.

[5] P. Flynn and A. Jain. CAD-based computer vision: from CAD models to relational graphs. *Transactions on Pattern Analysis and Machine Intelligence*, 13:114 – 132, 1991.

[6] M. Fritz and B. Schiele. Towards unsupervised discovery of visual categories. In *DAGM Symp. on Pattern Recog.*, 2006.

[7] W. Grimson, T. Lozano-Perez, and D. Huttenlocher. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, 1990.

[8] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Noelle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13:331 – 356, 1994.

[9] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3D object recognition. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[10] B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive mean-shift search. In *DAGM Symp. on Pattern Recognition*, 2004.

[11] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *Transactions on Pattern Analysis and Machine Intelligence*, 28:1465–1479, 2006.

[12] J. Liebelt and K. Schertler. Precise registration of 3D models to images by swarming particles. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[13] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355 – 395, March 1987.

[14] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. In *Conference on Computer Vision and Pattern Recognition*, 2006.

[15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence*, 27:1615 – 1630, 2005.

[16] H. Najafi, Y. Genc, and N. Navab. Fusion of 3D and appearance models for fast object detection and pose estimation. In *Asian Conference on Computer Vision*, 2006.

[17] C. F. Olson. Probabilistic indexing for object recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 17:518–522, 1995.

[18] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 66:231–259, 2006.

[19] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *International Conference on Computer Vision*, 2007.

[20] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision Graphics and Image Processing*, 40:361–387, 1987.

[21] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *Conf. on Computer Vision and Pattern Recognition*, 2006.

[22] M. Weber, W. Einhaeuser, M. Welling, and P. Perona. Viewpoint-invariant learning and detection of human heads. In *International Conference on Automatic Face and Gesture Recognition*, 2000.

[23] H. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *Comp. Science and Engineering*, 4:10–21, 1997.

[24] P. Yan, S. M. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *International Conference on Computer Vision*, 2007.